

**1. Copyright.**

Copyright © Dave Bone 1998 - 2015

**2. identifier thread.**

Does three things:

- 1) build up strings starting with letters following by alphanumerics, underscore and hyphens
- 2) determines whether its an identifier, keyword, or xxx-in-stbl where xxx one of rule, T.
- 3) it watches for the possibility of a shortened identifier within the lookahead expression.

This is caused by the hyphen being overloaded: within the lookahead expression it is an operator. The default is the longest possible identifier string which could be a keyword. Barring this it is an identifier up to but excluding the first hyphen. This is not pure in its error checking as the string could be in error within the lookahead expression and the faulty keyword being stated checked to the shortened identifier. Well the undefined identifier should be caught when the lookahead expression is postfixed evaluated against the symbol table.

To keep each drawing contained to a page, i broke the rules having many subrules like RUPPER into smaller rules: RUPPER\_A\_M and RUPPER\_N\_Z.

An Optimization:

Bypass the PDA of the grammar and use a FA for the tail-gating characters. Dtrace exposes well optimization opportunities. 2) Optimize: rid use of string to build up symbol name: use a raw buffer. Dtrace showed its inefficiencies: strlen, memcpy, and mutex locks.

Nov. 2008.

**3. Fsm Cidentifier class.****4. Cidentifier constructor directive.**

⟨ Cidentifier constructor directive 4 ⟩ ≡

```
hyphen_pos_ = -1;
hyphen_ = 0;
hyphen_idx_ = -1;
ddd_idx_ = 0;
ddd_[ddd_idx_] = 0;
```

**5. Cidentifier op directive.**

⟨ Cidentifier op directive 5 ⟩ ≡

```
hyphen_pos_ = -1;
hyphen_ = 0;
hyphen_idx_ = -1;
ddd_idx_ = 0;
ddd_[ddd_idx_] = 0;
```

**6. Cidentifier user-declaration directive.**

⟨ Cidentifier user-declaration directive 6 ⟩ ≡

```
public: int hyphen_pos_;
int hyphen_idx_;
CAbs_lr1_sym * hyphen_;
char ddd_[1024];
int ddd_idx_;
```

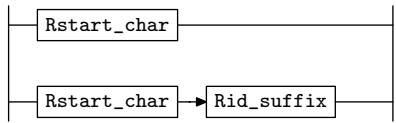
**7. Cidentifier user-prefix-declaration directive.**

⟨ Cidentifier user-prefix-declaration directive 7 ⟩ ≡

```
#include "yacco2_stbl.h"
```

**8. *Ridentifier* rule.**

*Ridentifier*



**9. Ridentifier op directive.**

Check if the identifier is a keyword by use of the symbol table and if so create the keyword and return it. Now return as an identifier with the prefix not containing the hyphen.

```

⟨Ridentifier op directive 9⟩ ≡
  using namespace NS_yacco2_T_enum;
  using namespace NS_yacco2_terminals;
  using namespace yacco2_stbl; Cidentifier * fsm = ( Cidentifier * ) rule_info_.parser_>fsm_tbl_;
  CAbs_lr1_sym * sym(0);
  T_sym_tbl_report_card report_card;
  find_sym_in_stbl(report_card, *fsm->ddd_);
  if (report_card.action_ ≡ T_sym_tbl_report_card::fnd) goto item_in_stbl;
  /* strip out - suffix as in a-b where suffix -b is bktracked */
  if (fsm->hyphen_ ≠ 0) { /* re-align to - as lookahead */
    rule_info_.parser_>override_current_token(*fsm->hyphen_, fsm->hyphen_pos_);
    fsm->ddd_[fsm->hyphen_idx_] = 0;
  }
  else {
    sym = new T_identifier((const char *) &fsm->ddd_);
    sym->set_rc(*rule_info_.parser_>start_token_, __FILE__, __LINE__);
    RSVP(sym);
    return;
  }
  find_sym_in_stbl(report_card, *fsm->ddd_); /* relook up id without "-" sufx */
  if (report_card.action_ ≡ T_sym_tbl_report_card::fnd) goto item_in_stbl;
  sym = new T_identifier((const char *) &fsm->ddd_);
  sym->set_rc(*rule_info_.parser_>start_token_, __FILE__, __LINE__);
  RSVP(sym);
  return;
item_in_stbl:
  if (report_card.tbl_entry->type_ ≠ table_entry::keyword) {
    /* return xxx-in-stbl where xxx one of rule, T */
    sym = report_card.tbl_entry->symbol_;
    sym->set_rc(*rule_info_.parser_>start_token_, __FILE__, __LINE__);
    RSVP(sym);
    return;
  }
  kw_handling: kw_in_stbl * kw_in = ( kw_in_stbl * ) report_card.tbl_entry->symbol_;
  CAbs_lr1_sym * kw = kw_in->keyword_in_stbl();
  CAbs_lr1_sym * nkwl;
  switch (kw->enumerated_id_) {
  case T_Enum::T_T_raw_characters_:
    {
      nkwl = new T_raw_characters;
      break;
    }
  case T_Enum::T_T_lr1_constant_symbols_:
    {
      nkwl = new T_lr1_constant_symbols;
      break;
    }
  case T_Enum::T_T_error_symbols_:
    {

```

```
    nkw = new T_error_symbols;  
    break;  
}  
case T_Enum::T_T_eocode_  
{  
    nkw = new T_eocode;  
    break;  
}  
case T_Enum::T_T_AD_  
{  
    nkw = new T_AD;  
    break;  
}  
case T_Enum::T_T_AB_  
{  
    nkw = new T_AB;  
    break;  
}  
case T_Enum::T_T_parallel_la_boundary_  
{  
    nkw = new T_parallel_la_boundary;  
    break;  
}  
case T_Enum::T_T_arbitrator_code_  
{  
    nkw = new T_arbitrator_code;  
    break;  
}  
case T_Enum::T_T_parallel_parser_  
{  
    nkw = new T_parallel_parser;  
    break;  
}  
case T_Enum::T_T_parallel_thread_function_  
{  
    nkw = new T_parallel_thread_function;  
    break;  
}  
case T_Enum::T_T_parallel_control_monitor_  
{  
    nkw = new T_parallel_control_monitor;  
    break;  
}  
case T_Enum::T_T_fsm_  
{  
    nkw = new T_fsm;  
    break;  
}  
case T_Enum::T_T_fsm_id_  
{  
    nkw = new T_fsm_id;  
    break;  
}
```

```

}
case T_Enum::T_T_fsm_filename_:
{
    nkw = new T_fsm_filename;
    break;
}
case T_Enum::T_T_fsm_namespace_:
{
    nkw = new T_fsm_namespace;
    break;
}
case T_Enum::T_T_fsm_class_:
{
    nkw = new T_fsm_class;
    break;
}
case T_Enum::T_T_fsm_version_:
{
    nkw = new T_fsm_version;
    break;
}
case T_Enum::T_T_fsm_date_:
{
    nkw = new T_fsm_date;
    break;
}
case T_Enum::T_T_fsm_debug_:
{
    nkw = new T_fsm_debug;
    break;
}
case T_Enum::T_T_fsm_comments_:
{
    nkw = new T_fsm_comments;
    break;
}
case T_Enum::T_T_terminals_:
{
    nkw = new T_terminals;
    break;
}
case T_Enum::T_T_enumeration_:
{
    nkw = new T_enumeration;
    break;
}
case T_Enum::T_T_file_name_:
{
    nkw = new T_file_name;
    break;
}
case T_Enum::T_T_name_space_:

```

```
{
    nkw = new T_name_space;
    break;
}
case T_Enum::T_T_sym_class_:
{
    nkw = new T_sym_class;
    break;
}
case T_Enum::T_T_rules_:
{
    nkw = new T_rules;
    break;
}
case T_Enum::T_T_lhs_:
{
    nkw = new T_lhs;
    break;
}
case T_Enum::T_T_user_declaration_:
{
    nkw = new T_user_declaration;
    break;
}
case T_Enum::T_T_user_prefix_declaration_:
{
    nkw = new T_user_prefix_declaration;
    break;
}
case T_Enum::T_T_user_suffix_declaration_:
{
    nkw = new T_user_suffix_declaration;
    break;
}
case T_Enum::T_T_constructor_:
{
    nkw = new T_constructor;
    break;
}
case T_Enum::T_T_destructor_:
{
    nkw = new T_destructor;
    break;
}
case T_Enum::T_T_op_:
{
    nkw = new T_op;
    break;
}
case T_Enum::T_T_failed_:
{
    nkw = new T_failed;
}
```

```

    break;
}
case T_Enum::T_T_user_implementation_:
{
    nkw = new T_user_implementation;
    break;
}
case T_Enum::T_T_user_imp_tbl_:
{
    nkw = new T_user_imp_tbl;
    break;
}
case T_Enum::T_T_user_imp_sym_:
{
    nkw = new T_user_imp_sym;
    break;
}
case T_Enum::T_T_constant_defs_:
{
    nkw = new T_constant_defs;
    break;
}
case T_Enum::T_T_terminals_refs_:
{
    nkw = new T_terminals_refs;
    break;
}
case T_Enum::T_T_terminals_sufx_:
{
    nkw = new T_terminals_sufx;
    break;
}
case T_Enum::T_T_lrk_sufx_:
{
    nkw = new T_lrk_sufx;
    break;
}
case T_Enum::T_LR1_eog_:
{
    nkw = new LR1_eog;
    break;
}
case T_Enum::T_LR1_eolr_:
{
    nkw = new LR1_eolr;
    break;
}
case T_Enum::T_T_NULL_:
{
    nkw = new T_NULL;
    break;
}
}

```



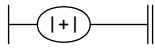
```

}
nkw→set_rc(*rule_info→parser→start_token→, __FILE__, __LINE__);
RSVP(nkw);
return;

```

### 10. *Rid\_suffix* rule.

Rid\_suffix



Pure finite automata for identifier suffix.

Capture the index of where the first hyphen is. As length is relative to one, this is its index before i add it to the being built up string. i could have added the character before and then used the length minus one to get its index: ahh the off by one count as the array operator is of course relative to zero. Dave u and your rants...and rolls.

Note the 1st entry character into this rule is on the stack which is what the stack frame uses to set the specific symbols. The current token for the parser is now the lookahead character. This is why I reset the parsing character upon entry to the previous token.

⟨Rid\_suffix subrule 1 op directive 10⟩ ≡

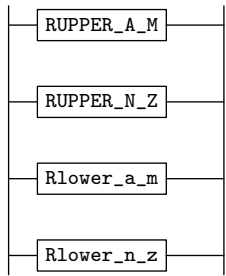
```

Cidentifier * fsm = ( Cidentifier * ) rule_info→parser→fsm_tbl→;
CAbs_lr1_sym * sym = sf→p1→;
parser()→reset_current_token(parser()→current_token_pos() - 1);
/* bk trk to previous chr which is on stack */
int id;
filter_char: id = sym→enumerated_id();
if ((id ≥ NS_yacco2_T_enum::T_Enum::T_raw_a_)^(id ≤ NS_yacco2_T_enum::T_Enum::T_raw_z_))
    goto bld_str;
if ((id ≥ NS_yacco2_T_enum::T_Enum::T_raw_A_) ^ (id ≤
    NS_yacco2_T_enum::T_Enum::T_raw_Z_)) goto bld_str;
if ((id ≥ NS_yacco2_T_enum::T_Enum::T_raw_0_)^(id ≤ NS_yacco2_T_enum::T_Enum::T_raw_9_))
    goto bld_str;
if (id ≡ NS_yacco2_T_enum::T_Enum::T_raw_under_score_) goto bld_str;
if (id ≡ NS_yacco2_T_enum::T_Enum::T_raw_minus_) {
    if (fsm→hyphen_ ≡ 0) {
        fsm→hyphen_pos_ = rule_info→parser→current_token_pos_ - 1;
        fsm→hyphen_ = sym;
        /* note: len rel 1, idx rel 0: "-" not added yet to str, this will be its idx value */
        fsm→hyphen_idx_ = fsm→ddd_idx_;
    }
    goto bld_str;
}
return; /* end-of-identifier */
bld_str: fsm→ddd_[fsm→ddd_idx_] = sym→id_[0];
++fsm→ddd_idx_;
fsm→ddd_[fsm→ddd_idx_] = 0;
parser()→get_next_token(); /* as current token */
sym = parser()→current_token→;
goto filter_char;

```

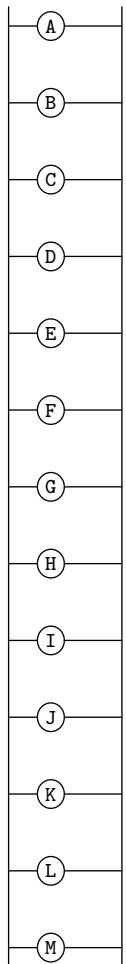
11. *Rstart\_char* rule.

*Rstart\_char*



12. RUPPER\_A\_M rule.

RUPPER\_A\_M



**13. RUPPER\_A\_M op directive.**

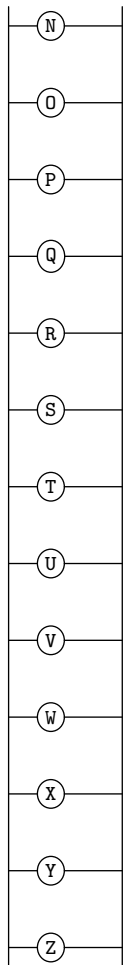
```

⟨ RUPPER_A_M op directive 13 ⟩ ≡
  Cidentifier * fsm = ( Cidentifier * ) rule_info...parser...fsm.tbl...;
  size_t pos = rule_info...parser...parse_stack...top_sub... - 1;
  CAbs_lr1_sym * sym = rule_info...parser...get_spec_stack_token(pos);
  fsm-ddd_[fsm-ddd_idx_] = sym-id_[0];
  ++fsm-ddd_idx_;
  fsm-ddd_[fsm-ddd_idx_] = 0;

```

**14. RUPPER\_N\_Z rule.**

RUPPER\_N\_Z



**15. RUPPER\_N\_Z op directive.**

⟨RUPPER\_N\_Z op directive 15⟩ ≡

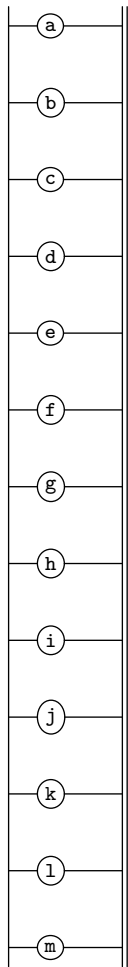
```

Cidentifier * fsm = ( Cidentifier * ) rule_info....parser...-fsm.tbl...;
size_t pos = rule_info....parser...-parse_stack....top_sub... - 1;
CAbs_lr1_sym * sym = rule_info....parser...-get_spec_stack_token(pos);
fsm-ddd_[fsm-ddd_idx_] = sym-id_[0];
++fsm-ddd_idx_;
fsm-ddd_[fsm-ddd_idx_] = 0;

```

**16. *Rlower\_a\_m* rule.**

*Rlower\_a\_m*



**17. Rlower\_a\_m op directive.**

⟨Rlower\_a\_m op directive 17⟩ ≡

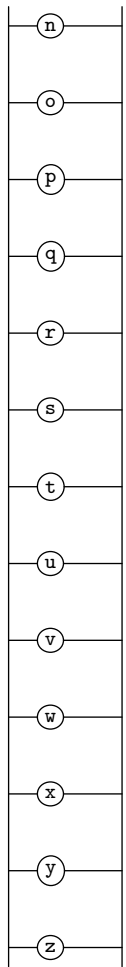
```

Cidentifier * fsm = ( Cidentifier * ) rule_info....parser...-fsm.tbl...;
size_t pos = rule_info....parser...-parse_stack....top_sub... - 1;
CAbs_lr1_sym * sym = rule_info....parser...-get_spec_stack_token(pos);
fsm-ddd_[fsm-ddd_idx_] = sym-id_[0];
++fsm-ddd_idx_;
fsm-ddd_[fsm-ddd_idx_] = 0;

```

**18. Rlower\_n\_z rule.**

Rlower\_n\_z



**19. Rlower\_n\_z op directive.**

⟨Rlower\_n\_z op directive 19⟩ ≡

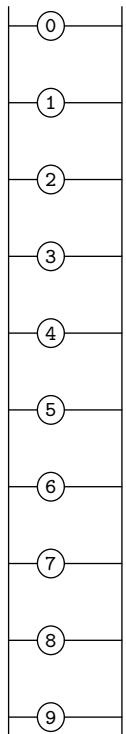
```

Cidentifier * fsm = ( Cidentifier * ) rule_info....parser...-fsm_tbl...;
size_t pos = rule_info....parser...-parse_stack....top_sub... - 1;
CAbs_lr1_sym * sym = rule_info....parser...-get_spec_stack_token(pos);
fsm-ddd_[fsm-ddd_idx_] = sym-id_[0];
++fsm-ddd_idx_;
fsm-ddd_[fsm-ddd_idx_] = 0;

```

**20. RNUMBERS rule.**

RNUMBERS

**21. RNUMBERS op directive.**

⟨RNUMBERS op directive 21⟩ ≡

```

Cidentifier * fsm = ( Cidentifier * ) rule_info....parser...-fsm_tbl...;
size_t pos = rule_info....parser...-parse_stack....top_sub... - 1;
CAbs_lr1_sym * sym = rule_info....parser...-get_spec_stack_token(pos);
fsm-ddd_[fsm-ddd_idx_] = sym-id_[0];
++fsm-ddd_idx_;
fsm-ddd_[fsm-ddd_idx_] = 0;

```

**22. First Set Language for  $O_2^{linker}$ .**

```
/*
  File: identifier.fsc
  Date and Time: Fri Jan  2 15:33:38 2015
*/
transitive      n
grammar-name    "identifier"
name-space      "NS_identifier"
thread-name     "TH_identifier"
monolithic      n
file-name       "identifier.fsc"
no-of-T         569
list-of-native-first-set-terminals 52
  raw_A
  raw_B
  raw_C
  raw_D
  raw_E
  raw_F
  raw_G
  raw_H
  raw_I
  raw_J
  raw_K
  raw_L
  raw_M
  raw_N
  raw_O
  raw_P
  raw_Q
  raw_R
  raw_S
  raw_T
  raw_U
  raw_V
  raw_W
  raw_X
  raw_Y
  raw_Z
  raw_a
  raw_b
  raw_c
  raw_d
  raw_e
  raw_f
  raw_g
  raw_h
  raw_i
  raw_j
  raw_k
  raw_l
```

```
raw_m
raw_n
raw_o
raw_p
raw_q
raw_r
raw_s
raw_t
raw_u
raw_v
raw_w
raw_x
raw_y
raw_z
end-list-of-native-first-set-terminals
list-of-transitive-threads 0
end-list-of-transitive-threads
list-of-used-threads 0
end-list-of-used-threads
fsm-comments
"Yacco2 identifiers lexer with symbol table lookup."
```



**23. Lr1 State Network.**

⇒

State: 1 state type: <sup>s</sup>

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	RUPPER_A.M	4	1	1	A			1	2	2		
c	RUPPER_A.M	4	2	1	B			1	3	3		
c	RUPPER_A.M	4	3	1	C			1	4	4		
c	RUPPER_A.M	4	4	1	D			1	5	5		
c	RUPPER_A.M	4	5	1	E			1	6	6		
c	RUPPER_A.M	4	6	1	F			1	7	7		
c	RUPPER_A.M	4	7	1	G			1	8	8		
c	RUPPER_A.M	4	8	1	H			1	9	9		
c	RUPPER_A.M	4	9	1	I			1	10	10		
c	RUPPER_A.M	4	10	1	J			1	11	11		
c	RUPPER_A.M	4	11	1	K			1	12	12		
c	RUPPER_A.M	4	12	1	L			1	13	13		
c	RUPPER_A.M	4	13	1	M			1	14	14		
c	RUPPER_N.Z	5	1	1	N			1	15	15		
c	RUPPER_N.Z	5	2	1	O			1	16	16		
c	RUPPER_N.Z	5	3	1	P			1	17	17		
c	RUPPER_N.Z	5	4	1	Q			1	18	18		
c	RUPPER_N.Z	5	5	1	R			1	19	19		
c	RUPPER_N.Z	5	6	1	S			1	20	20		
c	RUPPER_N.Z	5	7	1	T			1	21	21		
c	RUPPER_N.Z	5	8	1	U			1	22	22		
c	RUPPER_N.Z	5	9	1	V			1	23	23		
c	RUPPER_N.Z	5	10	1	W			1	24	24		
c	RUPPER_N.Z	5	11	1	X			1	25	25		
c	RUPPER_N.Z	5	12	1	Y			1	26	26		
c	RUPPER_N.Z	5	13	1	Z			1	27	27		
c	Rlower_a.m	6	1	1	a			1	28	28		
c	Rlower_a.m	6	2	1	b			1	29	29		
c	Rlower_a.m	6	3	1	c			1	30	30		
c	Rlower_a.m	6	4	1	d			1	31	31		
c	Rlower_a.m	6	5	1	e			1	32	32		
c	Rlower_a.m	6	6	1	f			1	33	33		
c	Rlower_a.m	6	7	1	g			1	34	34		
c	Rlower_a.m	6	8	1	h			1	35	35		
c	Rlower_a.m	6	9	1	i			1	36	36		
c	Rlower_a.m	6	10	1	j			1	37	37		
c	Rlower_a.m	6	11	1	k			1	38	38		
c	Rlower_a.m	6	12	1	l			1	39	39		
c	Rlower_a.m	6	13	1	m			1	40	40		
c	Rlower_n.z	7	1	1	n			1	41	41		
c	Rlower_n.z	7	2	1	o			1	42	42		
c	Rlower_n.z	7	3	1	p			1	43	43		
c	Rlower_n.z	7	4	1	q			1	44	44		
c	Rlower_n.z	7	5	1	r			1	45	45		
c	Rlower_n.z	7	6	1	s			1	46	46		
c	Rlower_n.z	7	7	1	t			1	47	47		
c	Rlower_n.z	7	8	1	u			1	48	48		
c	Rlower_n.z	7	9	1	v			1	49	49		

c Rlower_n_z	7	10	1	w		1	50	50	
c Rlower_n_z	7	11	1	x		1	51	51	
c Rlower_n_z	7	12	1	y		1	52	52	
c Rlower_n_z	7	13	1	z		1	53	53	
c Ridentifier	1	2	1	Rstart_char <u>Rid_suffix</u>		1	54	56	
c Ridentifier	1	1	1	Rstart_char		1	54	54	
c Rstart_char	3	1	1	RUPPER_A_M		1	57	57	
c Rstart_char	3	2	1	RUPPER_N_Z		1	58	58	
c Rstart_char	3	3	1	Rlower_a_m		1	59	59	
c Rstart_char	3	4	1	Rlower_n_z		1	60	60	
$\Rightarrow^A$									
← rule	→ R#	sr#	Po	←	State: 2 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	1	2		subrule element	1	0	2	1
$\Rightarrow^B$									
← rule	→ R#	sr#	Po	←	State: 3 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	2	2		subrule element	1	0	3	1
$\Rightarrow^C$									
← rule	→ R#	sr#	Po	←	State: 4 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	3	2		subrule element	1	0	4	1
$\Rightarrow^D$									
← rule	→ R#	sr#	Po	←	State: 5 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	4	2		subrule element	1	0	5	1
$\Rightarrow^E$									
← rule	→ R#	sr#	Po	←	State: 6 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	5	2		subrule element	1	0	6	1
$\Rightarrow^F$									
← rule	→ R#	sr#	Po	←	State: 7 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	6	2		subrule element	1	0	7	1
$\Rightarrow^G$									
← rule	→ R#	sr#	Po	←	State: 8 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	7	2		subrule element	1	0	8	1
$\Rightarrow^H$									
← rule	→ R#	sr#	Po	←	State: 9 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	8	2		subrule element	1	0	9	1
$\Rightarrow^I$									
← rule	→ R#	sr#	Po	←	State: 10 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	9	2		subrule element	1	0	10	1
$\Rightarrow^J$									
← rule	→ R#	sr#	Po	←	State: 11 state type: <i>r</i>	→ Brn	Gto	Red	LA
t RUPPER_A_M	4	10	2		subrule element	1	0	11	1
$\Rightarrow^K$									
					State: 12 state type: <i>r</i>				

← rule t RUPPER_A_M	→ R# sr# Po ← 4 11 2	← subrule element	→ Brn Gto Red LA 1 0 12 1
⇒ <sup>L</sup>		State: 13 state type: <sup>r</sup>	
← rule t RUPPER_A_M	→ R# sr# Po ← 4 12 2	← subrule element	→ Brn Gto Red LA 1 0 13 1
⇒ <sup>M</sup>		State: 14 state type: <sup>r</sup>	
← rule t RUPPER_A_M	→ R# sr# Po ← 4 13 2	← subrule element	→ Brn Gto Red LA 1 0 14 1
⇒ <sup>N</sup>		State: 15 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 1 2	← subrule element	→ Brn Gto Red LA 1 0 15 1
⇒ <sup>O</sup>		State: 16 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 2 2	← subrule element	→ Brn Gto Red LA 1 0 16 1
⇒ <sup>P</sup>		State: 17 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 3 2	← subrule element	→ Brn Gto Red LA 1 0 17 1
⇒ <sup>Q</sup>		State: 18 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 4 2	← subrule element	→ Brn Gto Red LA 1 0 18 1
⇒ <sup>R</sup>		State: 19 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 5 2	← subrule element	→ Brn Gto Red LA 1 0 19 1
⇒ <sup>S</sup>		State: 20 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 6 2	← subrule element	→ Brn Gto Red LA 1 0 20 1
⇒ <sup>T</sup>		State: 21 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 7 2	← subrule element	→ Brn Gto Red LA 1 0 21 1
⇒ <sup>U</sup>		State: 22 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 8 2	← subrule element	→ Brn Gto Red LA 1 0 22 1
⇒ <sup>V</sup>		State: 23 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 9 2	← subrule element	→ Brn Gto Red LA 1 0 23 1
⇒ <sup>W</sup>		State: 24 state type: <sup>r</sup>	
← rule t RUPPER_N_Z	→ R# sr# Po ← 5 10 2	← subrule element	→ Brn Gto Red LA 1 0 24 1
⇒ <sup>X</sup>		State: 25 state type: <sup>r</sup>	

← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t RUPPER_N_Z	5 11 2		1 0 25 1
⇒ <sup>Y</sup>		State: 26 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t RUPPER_N_Z	5 12 2		1 0 26 1
⇒ <sup>Z</sup>		State: 27 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t RUPPER_N_Z	5 13 2		1 0 27 1
⇒ <sup>a</sup>		State: 28 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 1 2		1 0 28 1
⇒ <sup>b</sup>		State: 29 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 2 2		1 0 29 1
⇒ <sup>c</sup>		State: 30 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 3 2		1 0 30 1
⇒ <sup>d</sup>		State: 31 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 4 2		1 0 31 1
⇒ <sup>e</sup>		State: 32 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 5 2		1 0 32 1
⇒ <sup>f</sup>		State: 33 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 6 2		1 0 33 1
⇒ <sup>g</sup>		State: 34 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 7 2		1 0 34 1
⇒ <sup>h</sup>		State: 35 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 8 2		1 0 35 1
⇒ <sup>i</sup>		State: 36 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 9 2		1 0 36 1
⇒ <sup>j</sup>		State: 37 state type: <i>r</i>	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rlower_a_m	6 10 2		1 0 37 1
⇒ <sup>k</sup>		State: 38 state type: <i>r</i>	

← rule t Rlower_a_m	→ R# sr# Po ← 6 11 2	← subrule element	→ Brn Gto Red LA 1 0 38 1
⇒ <sup>l</sup>		State: 39 state type: <sup>r</sup>	
← rule t Rlower_a_m	→ R# sr# Po ← 6 12 2	← subrule element	→ Brn Gto Red LA 1 0 39 1
⇒ <sup>m</sup>		State: 40 state type: <sup>r</sup>	
← rule t Rlower_a_m	→ R# sr# Po ← 6 13 2	← subrule element	→ Brn Gto Red LA 1 0 40 1
⇒ <sup>n</sup>		State: 41 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 1 2	← subrule element	→ Brn Gto Red LA 1 0 41 1
⇒ <sup>o</sup>		State: 42 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 2 2	← subrule element	→ Brn Gto Red LA 1 0 42 1
⇒ <sup>p</sup>		State: 43 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 3 2	← subrule element	→ Brn Gto Red LA 1 0 43 1
⇒ <sup>q</sup>		State: 44 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 4 2	← subrule element	→ Brn Gto Red LA 1 0 44 1
⇒ <sup>r</sup>		State: 45 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 5 2	← subrule element	→ Brn Gto Red LA 1 0 45 1
⇒ <sup>s</sup>		State: 46 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 6 2	← subrule element	→ Brn Gto Red LA 1 0 46 1
⇒ <sup>t</sup>		State: 47 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 7 2	← subrule element	→ Brn Gto Red LA 1 0 47 1
⇒ <sup>u</sup>		State: 48 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 8 2	← subrule element	→ Brn Gto Red LA 1 0 48 1
⇒ <sup>v</sup>		State: 49 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 9 2	← subrule element	→ Brn Gto Red LA 1 0 49 1
⇒ <sup>w</sup>		State: 50 state type: <sup>r</sup>	
← rule t Rlower_n_z	→ R# sr# Po ← 7 10 2	← subrule element	→ Brn Gto Red LA 1 0 50 1
⇒ <sup>x</sup>		State: 51 state type: <sup>r</sup>	

$\leftarrow$ rule t Rlower_n_z	$\rightarrow$ R# sr# Po $\leftarrow$ 7 11 2	subrule element	$\rightarrow$ Brn Gto Red LA 1 0 51 1
$\Rightarrow^y$			
$\leftarrow$ rule t Rlower_n_z	$\rightarrow$ R# sr# Po $\leftarrow$ 7 12 2	State: 52 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 52 1
$\Rightarrow^z$			
$\leftarrow$ rule t Rlower_n_z	$\rightarrow$ R# sr# Po $\leftarrow$ 7 13 2	State: 53 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 53 1
$\Rightarrow^{Rstart\_char}$			
$\leftarrow$ rule t Ridentifier c Rid_suffix t Ridentifier	$\rightarrow$ R# sr# Po $\leftarrow$ 1 1 2 2 1 1  +  1 2 2 Rid_suffix	State: 54 state type: $s/r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 54 1 54 55 55 1 56 56
$\Rightarrow^{+ }$			
$\leftarrow$ rule t Rid_suffix	$\rightarrow$ R# sr# Po $\leftarrow$ 2 1 2	State: 55 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 54 0 55 1
$\Rightarrow^{Rid\_suffix}$			
$\leftarrow$ rule t Ridentifier	$\rightarrow$ R# sr# Po $\leftarrow$ 1 2 3	State: 56 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 56 1
$\Rightarrow^{RUPPER\_A\_M}$			
$\leftarrow$ rule t Rstart_char	$\rightarrow$ R# sr# Po $\leftarrow$ 3 1 2	State: 57 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 57 1
$\Rightarrow^{RUPPER\_N\_Z}$			
$\leftarrow$ rule t Rstart_char	$\rightarrow$ R# sr# Po $\leftarrow$ 3 2 2	State: 58 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 58 1
$\Rightarrow^{Rlower\_a\_m}$			
$\leftarrow$ rule t Rstart_char	$\rightarrow$ R# sr# Po $\leftarrow$ 3 3 2	State: 59 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 59 1
$\Rightarrow^{Rlower\_n\_z}$			
$\leftarrow$ rule t Rstart_char	$\rightarrow$ R# sr# Po $\leftarrow$ 3 4 2	State: 60 state type: $r$ subrule element	$\rightarrow$ Brn Gto Red LA 1 0 60 1

**24. Index.**

- |+|: 10.
- \_\_FILE\_\_: 9.
- \_\_LINE\_\_: 9.
- action\_: 9.
- bld\_str: 10.
- CAbs\_lr1\_sym: 6, 9, 10, 13, 15, 17, 19, 21.
- Cidentifier: 9, 10, 13, 15, 17, 19, 21.
- current\_token\_\_: 10.
- current\_token\_pos: 10.
- current\_token\_pos\_\_: 10.
- ddd\_: 4, 5, 6, 9, 10, 13, 15, 17, 19, 21.
- ddd\_idx\_: 4, 5, 6, 10, 13, 15, 17, 19, 21.
- enumerated\_id: 10.
- enumerated\_id\_\_: 9.
- filter\_char: 10.
- find\_sym\_in\_stbl: 9.
- fnd: 9.
- fsm: 9, 10, 13, 15, 17, 19, 21.
- fsm\_tbl\_\_: 9, 10, 13, 15, 17, 19, 21.
- get\_next\_token: 10.
- get\_spec\_stack\_token: 13, 15, 17, 19, 21.
- hyphen\_: 4, 5, 6, 9, 10.
- hyphen\_idx\_: 4, 5, 6, 9, 10.
- hyphen\_pos\_: 4, 5, 6, 9, 10.
- id: 10.
- id\_\_: 10, 13, 15, 17, 19, 21.
- identifier: 2.
- item\_in\_stbl: 9.
- keyword: 9.
- keyword\_in\_stbl: 9.
- kw: 9.
- kw\_handling: 9.
- kw\_in: 9.
- kw\_in\_stbl: 9.
- LR1\_eog: 9.
- LR1\_eolr: 9.
- nkw: 9.
- NS\_yacco2\_T\_enum: 9, 10.
- NS\_yacco2\_terminals: 9.
- override\_current\_token: 9.
- parse\_stack\_\_: 13, 15, 17, 19, 21.
- parser: 10.
- parser\_\_: 9, 10, 13, 15, 17, 19, 21.
- pos: 13, 15, 17, 19, 21.
- p1\_\_: 10.
- report\_card: 9.
- reset\_current\_token: 10.
- Rid\_suffix: 8.
- Rid\_suffix: 10.
- Ridentifier: 8.
- Rlower\_a\_m: 11.
- Rlower\_n\_z: 11.
- Rlower\_a\_m: 16.
- Rlower\_n\_z: 18.
- RNUMBERS: 20.
- Rstart\_char: 8.
- Rstart\_char: 11.
- RSVP: 9.
- rule\_info\_\_: 9, 10, 13, 15, 17, 19, 21.
- RUPPER: 2.
- RUPPER\_A\_M: 11.
- RUPPER\_N\_Z: 11.
- RUPPER\_A\_M: 2, 12.
- RUPPER\_N\_Z: 2, 14.
- set\_rc: 9.
- sf: 10.
- start\_token\_\_: 9.
- sym: 9, 10, 13, 15, 17, 19, 21.
- symbol\_: 9.
- T\_AB: 9.
- T\_AD: 9.
- T\_arbitrator\_code: 9.
- T\_constant\_defs: 9.
- T\_constructor: 9.
- T\_destructor: 9.
- T\_Enum: 9, 10.
- T\_enumeration: 9.
- T\_eocode: 9.
- T\_error\_symbols: 9.
- T\_failed: 9.
- T\_file\_name: 9.
- T\_fsm: 9.
- T\_fsm\_class: 9.
- T\_fsm\_comments: 9.
- T\_fsm\_date: 9.
- T\_fsm\_debug: 9.
- T\_fsm\_filename: 9.
- T\_fsm\_id: 9.
- T\_fsm\_namespace: 9.
- T\_fsm\_version: 9.
- T\_identifier: 9.
- T\_lhs: 9.
- T\_lrk\_sufx: 9.
- T\_lr1\_constant\_symbols: 9.
- T\_LR1\_eog\_: 9.
- T\_LR1\_eolr\_: 9.
- T\_name\_space: 9.
- T\_NULL: 9.
- T\_op: 9.
- T\_parallel\_control\_monitor: 9.
- T\_parallel\_la\_boundary: 9.
- T\_parallel\_parser: 9.

- T\_parallel\_thread\_function*: 9.
- T\_raw\_a*: 10.
- T\_raw\_A*: 10.
- T\_raw\_characters*: 9.
- T\_raw\_minus*: 10.
- T\_raw\_under\_score*: 10.
- T\_raw\_z*: 10.
- T\_raw\_Z*: 10.
- T\_raw\_0*: 10.
- T\_raw\_9*: 10.
- T\_rules*: 9.
- T\_sym\_class*: 9.
- T\_sym\_tbl\_report\_card*: 9.
- T\_T\_AB*: 9.
- T\_T\_AD*: 9.
- T\_T\_arbitrator\_code*: 9.
- T\_T\_constant\_defs*: 9.
- T\_T\_constructor*: 9.
- T\_T\_destructor*: 9.
- T\_T\_enumeration*: 9.
- T\_T\_eocode*: 9.
- T\_T\_error\_symbols*: 9.
- T\_T\_failed*: 9.
- T\_T\_file\_name*: 9.
- T\_T\_fsm*: 9.
- T\_T\_fsm\_class*: 9.
- T\_T\_fsm\_comments*: 9.
- T\_T\_fsm\_date*: 9.
- T\_T\_fsm\_debug*: 9.
- T\_T\_fsm\_filename*: 9.
- T\_T\_fsm\_id*: 9.
- T\_T\_fsm\_namespace*: 9.
- T\_T\_fsm\_version*: 9.
- T\_T\_lhs*: 9.
- T\_T\_lrk\_sufx*: 9.
- T\_T\_lr1\_constant\_symbols*: 9.
- T\_T\_name\_space*: 9.
- T\_T\_NULL*: 9.
- T\_T\_op*: 9.
- T\_T\_parallel\_control\_monitor*: 9.
- T\_T\_parallel\_la\_boundary*: 9.
- T\_T\_parallel\_parser*: 9.
- T\_T\_parallel\_thread\_function*: 9.
- T\_T\_raw\_characters*: 9.
- T\_T\_rules*: 9.
- T\_T\_sym\_class*: 9.
- T\_T\_terminals*: 9.
- T\_T\_terminals\_refs*: 9.
- T\_T\_terminals\_sufx*: 9.
- T\_T\_user\_declaration*: 9.
- T\_T\_user\_imp\_sym*: 9.
- T\_T\_user\_imp\_tbl*: 9.
- T\_T\_user\_implementation*: 9.
- T\_T\_user\_prefix\_declaration*: 9.
- T\_T\_user\_suffix\_declaration*: 9.
- T\_terminals*: 9.
- T\_terminals\_refs*: 9.
- T\_terminals\_sufx*: 9.
- T\_user\_declaration*: 9.
- T\_user\_imp\_sym*: 9.
- T\_user\_imp\_tbl*: 9.
- T\_user\_implementation*: 9.
- T\_user\_prefix\_declaration*: 9.
- T\_user\_suffix\_declaration*: 9.
- table\_entry*: 9.
- tbl\_entry*: 9.
- top\_sub*: 13, 15, 17, 19, 21.
- type*: 9.
- yacco2\_stbl**: [9](#).



- ⟨ Cidentifier constructor directive 4 ⟩
- ⟨ Cidentifier op directive 5 ⟩
- ⟨ Cidentifier user-declaration directive 6 ⟩
- ⟨ Cidentifier user-prefix-declaration directive 7 ⟩
- ⟨ RNUMBERS op directive 21 ⟩
- ⟨ RUPPER\_A\_M op directive 13 ⟩
- ⟨ RUPPER\_N\_Z op directive 15 ⟩
- ⟨ Rid\_suffix subrule 1 op directive 10 ⟩
- ⟨ Ridentifier op directive 9 ⟩
- ⟨ Rlower\_a\_m op directive 17 ⟩
- ⟨ Rlower\_n\_z op directive 19 ⟩

# identifier Grammar

Date: January 2, 2015 at 15:36

File: identifier.lex                      Ns: NS\_identifier

Version: 1.0                                Debug: false

Grammar Comments:                        Type: Thread

Yacco2 identifiers lexer with symbol table lookup.

1 element(s) in Lookahead Expression below

eor

	Section	Page
<b>Copyright</b> .....	1	1
<i>identifier</i> <b>thread</b> .....	2	2
Fsm Cidentifier class .....	3	2
Cidentifier constructor directive .....	4	2
Cidentifier op directive .....	5	2
Cidentifier user-declaration directive .....	6	2
Cidentifier user-prefix-declaration directive .....	7	2
<i>Ridentifier</i> rule .....	8	3
Ridentifier op directive .....	9	4
<i>Rid_suffix</i> rule .....	10	9
<i>Rstart_char</i> rule .....	11	10
RUPPER_A_M rule .....	12	10
RUPPER_A_M op directive .....	13	11
RUPPER_N_Z rule .....	14	11
RUPPER_N_Z op directive .....	15	12
<i>Rlower_a_m</i> rule .....	16	12
Rlower_a_m op directive .....	17	13
<i>Rlower_n_z</i> rule .....	18	13
Rlower_n_z op directive .....	19	14
RNUMBERS rule .....	20	14
RNUMBERS op directive .....	21	14
<b>First Set Language for <math>O_2^{linker}</math></b> .....	22	15
<b>Lr1 State Network</b> .....	23	17
<b>Index</b> .....	24	23